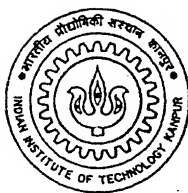


# Data Driven Feature Extraction and Parameterization for Speech Recognition

*A Thesis Submitted  
in Partial Fulfillment of the Requirements  
for the Degree of  
Master of Technology*

by  
Madhav Pandya




*to the*  
Department of Computer Science & Engineering  
Indian Institute of Technology, Kanpur

July, 2005

# Certificate

This is to certify that the work contained in the thesis entitled "*Data Driven Feature Extraction and Parameterization for Speech Recognition*", by *Madhav Pandya*, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

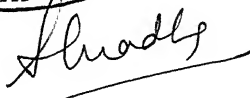
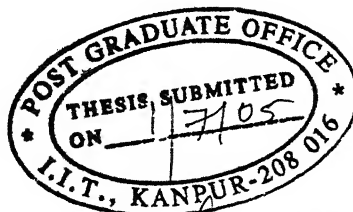
June, 2005



---

(Dr. Harish Karnick)

Department of Computer Science & Engineering,  
Indian Institute of Technology,  
Kanpur.



TH  
CSE/2005/M  
P192d

13 OCT 2005 (CSE)

पुस्तकालय  
भारत सरकार पुस्तकालय  
संस्थान कानपुर  
प्राप्ति क्र. A 153070



A153070

## Abstract

A pattern recognizer is usually a modular system, which consist of a feature extraction module and a recognition module. Traditionally, these two systems have been designed separately, which may not result in optimal recognition accuracy. Speech recognizer is one such system where the feature extraction module is designed using expert knowledge. In our approach, we apply a Genetic Algorithm to evolve a feature extractor based on the fitness evaluation for phonemically tagged data. A feature extractor can be evaluated based on its ability to map instances of different classes into different regions of the feature space. A measure of goodness can be associated with a feature extractor using class dissimilarity measures taking phonemes as classes. Different feature extractors can be produced by varying the boundaries of the filters of the filter-bank of a Mel Scale Frequency Cepstrum Coefficient(MFCC) like feature extractor. We use class separability measure as a fitness function and filter-bank as an evolving element to run a Genetic Algorithm based optimization. We have achieved significant accuracy increase on average using evolved feature extractor over MFCC.

# Acknowledgements

I take this immense opportunity to express my sincere gratitude my supervisor Dr. Harish Karnick for his guidance throughout the span of thesis. I consider myself extremely fortunate to have had a chance to work under his supervision. Learning how to do research was a very valuable experience for my growth as a researcher. In spite of his hectic schedule, he was able to give time for discussions providing guidance and invaluable suggestions for my research work. His interest and confidence in me was the reason for all my success. He was always ready to discuss new ideas and encouraged me to pursue them. It has been a very enlightening and enjoyable experience to work under him.

I would also like to thank the *Media Lab Asia* staff for collecting the speech which was very important for my experiments. I wish to thank Prologix Software Pvt. Ltd. for providing valuable tagged speech data.

I am thankful to all my friends in IITK who made my stay memorable. I would like to thank my parents and sister for providing constant support and encouragement to pursue my goals.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	2
1.2	Problem Definition . . . . .	3
1.3	Organization of thesis . . . . .	4
<b>2</b>	<b>Fundamentals of Speech Recognition</b>	<b>5</b>
2.1	Major Approaches . . . . .	6
2.2	Feature Extraction . . . . .	9
2.3	Classification . . . . .	16
2.3.1	Acoustic Modeling . . . . .	16
2.3.2	Language Modeling . . . . .	19
2.3.3	Decoding Algorithm . . . . .	21
<b>3</b>	<b>Previous Work and Preliminaries</b>	<b>23</b>
3.1	Optimization of Features for pattern recognition . . . . .	23
3.2	Genetic Algorithm as an Optimization Method . . . . .	27
3.2.1	Genetic Operators . . . . .	29
<b>4</b>	<b>Design of GA Based Feature Extraction Optimization System</b>	<b>32</b>
4.1	Genetic Representation . . . . .	33
4.2	Genetic Operators . . . . .	34
4.2.1	Initialization . . . . .	34
4.2.2	Mutation . . . . .	35
4.2.3	Crossover . . . . .	35

4.3	Objective Function . . . . .	36
4.3.1	Feature Extraction . . . . .	38
4.3.2	Feature Vector Sequence Length Normalization . . . . .	39
4.3.3	Class Separability Measurement . . . . .	42
<b>5</b>	<b>Experiments and Results</b>	<b>46</b>
5.1	Baseline Recognition System: SPHINX . . . . .	46
5.2	Data Sets . . . . .	48
5.3	Hindi Phoneme Recognition Task . . . . .	49
<b>6</b>	<b>Conclusion and Future Work</b>	<b>55</b>
6.1	Conclusion . . . . .	55
6.2	Future Work . . . . .	56
	<b>Bibliography</b>	<b>57</b>

# List of Tables

5.1	Parameter values for AllPhoneme Experiment . . . . .	51
5.2	Comparison of $FB_{mel}$ and $FB_{allphoneme}$ on <i>MLA</i> and <i>Hindi</i> data set .	52



# List of Figures

2.1	General Speech Recognition System . . . . .	8
2.2	Components Contributing to Speech Signal . . . . .	9
2.3	Role of Feature Extraction in Speech Recognition and Speaker Recognition . . . . .	10
2.4	Frame based feature extraction . . . . .	11
2.5	Filter-bank based feature extraction . . . . .	13
2.6	Mel frequency scale approximation proposed by Fant . . . . .	16
2.7	A three state left-to-right HMM . . . . .	17
2.8	Search graph generated by grammar . . . . .	20
3.1	Genetic Algorithm . . . . .	28
3.2	Crossover operation . . . . .	30
4.1	A filter bank . . . . .	34
4.2	Objective Function Evaluation . . . . .	36
4.3	Top: Original Signal Bottom: Reconstructed Signal . . . . .	41
4.4	Original and Reconstructed Signals . . . . .	42
4.5	Decorrelation in measurement space . . . . .	44
5.1	Sphinx - 4 Decoder Framework . . . . .	47
5.2	Sphinx-4 Front-end . . . . .	48
5.3	All Phoneme Experiment Results . . . . .	50
5.4	$FB_{allphoneme}$ (top) and $FB_{mel}$ (botton) . . . . .	52
5.5	Center Frequencies of the Best Filter Bank . . . . .	53
5.6	Vowel Experiment . . . . .	54

# Chapter 1

## Introduction

The goal of a speech recognition system is to accurately and efficiently convert a speech signal into a text message independent of the device, speaker or the environment.

Naturally, one of the most promising approaches to speech recognition is to investigate the human process of speech perception and to apply the scientific findings to the development of a recognition system. This human-science-based approach has not yet achieved comprehensive analysis of human mechanism, though the partial findings have been used in the current systems. The modular design of the system is also encouraged by the human-science-based findings. First a feature extraction module generates features from given speech signal and recognizer module uses these features to classify the input signal to output the recognized hypothesis.

The design of a feature extraction module is highly inspired by human perception process, while the recognizer is designed using different machine learning models. For feature extraction, a Mel-frequency cepstrum coefficient(MFCC) method is most widespread in such systems. Continuous research in this field for approximately four decades has taken the recognition accuracy nearly 100%. This progress can be largely rewarded to better speech modeling techniques. Today's state of the art systems contain HMM based models which have been found to be the best model for speech[18]. This is because of the fact that HMM can implicitly handle the

variations in the length.

Today, there are many speech recognition systems available in both research and commercial domain. SPHINX, HTK and ISIP are examples of open source systems. While Dragon Systems by ScanSoft and ViaVoice from IBM are examples of commercial system available.

## 1.1 Motivation

In spite of such a progress in recognition accuracy, automatic speech recognition (ASR) systems still have a long way to go before they can be used for routine input. The main difficulties arise due to:

- Change in the environment of operation
- High degree of noise
- Change in the channel characteristics
- Speaker variations
- Change in the microphone characteristics

Another problem with current systems is high memory and computational requirements. It is also important to note that these problems are correlated, for example decreasing the memory and computation requirements degrade robustness of the system. Such trade-offs compel us to fine tune the system for a particular application. Researchers are now focusing on improving the memory and computation aspects while maintaining system performance. There are many approaches a solution. One obvious direction would be to analyze the Human Speech Recognition(HSR) System. Studies show that the perception process is not only utilizing the acoustic and phonetics knowledge but also higher level of knowledge such as *semantics*, *context* and *pragmatics*. Unless we incorporate this information, we will not be able to attain HSR performance. If we carefully think of the content of the signal, it contains the following information.

- Speaker Information (SI)
- Signal Message (SM)
- Environmental Characteristics(EC)

One of the causes of the HRS lies in its ability to separate different information for different purposes. Like, listening speech segment, we would be able to say that it was a message “come here” from a person X and TV was playing in the background. If we are able to find such a mechanism to extract a piece of information relevant to the intent, it would be a big step towards making speech recognition a real working system. Work in the speech modeling domain has seen tremendous amount of work giving high accuracies and it seems it has reached a performance ceiling. Only better features for recognition can improve the situation any further. Hence our work is focused on studying this problem.

We need to find feature set with lower dimensionality and richest discriminant information. Decreasing the dimensionality while keeping the discriminant information would yield reduced memory and computation requirements. This is because the means and variances vectors takes less memory and computation of the probability of a frame with respect to a state reduces also.

## 1.2 Problem Definition

In this thesis, we have focused on the process of extraction of optimal features from the speech signal that would help the onward recognition process decide the correct hypothesis. The goodness of a feature extraction system is determined based on the feature space generated. A Genetic algorithm has been used as an optimization mechanism to obtain the best features for the task.

## 1.3 Organization of thesis

The organization of the thesis is as follows. Chapter 2 provides an introduction to speech recognition. It first provides the mathematical formulation of the problem and describes the problem as a Pattern Recognition task splitting the whole process into two main parts: Feature Extraction and Classification. Focus is given to various feature extraction methods where we describe the filter-bank based feature extraction in detail.

Chapter 3 contains some of the concepts used in our work and previous work done in this domain. These are Genetic Algorithm and Discriminative Feature Extraction. A general formulation of a genetic algorithm and its major components have been described. It also reviews the work done in the domain of feature extraction learning for speech recognition. It mainly consists of the kinds of feature extractors considered and methods to obtain the goodness estimation of the same.

Our formulation of the filter-bank optimization problem using a genetic algorithm is described in Chapter 4. Chapter 5 describes the experiments done and the results. Finally, chapter 6 concludes providing some future direction for further work.

## Chapter 2

# Fundamentals of Speech Recognition

The basic problem of speech recognition is to find the best word sequence corresponding to the speech signal. The word sequence comes from a set of possible word sequences, called the language  $L$ . Also the words are coming from a closed set called the dictionary  $D$ . The language can contain infinite number of sentences based on the grammar provided. The task of the recognizer is to find a valid word sequence present in the given language  $L$ . Speech is a continuous time analog signal. This kind of signal can not be directly processed by digital systems. Hence, sampling and quantization is performed to transform the input continuous speech signal into a discrete signal with quantized amplitude. A pre-processing system(also known as feature extractor or front end processing) transforms this into a sequence of feature vectors.

Broadly, we can divide the speech recognition area into two branches[5]:

- Isolated Word Recognition(*IWR*)
- Continuous Speech Recognition(*CSR*)

In IWR, the recognizer takes as input an observation sequence of any utterance at a time(spoken in isolation and belonging a fixed dictionary) and outputs the word which has the highest probability corresponding that observation sequence. In these systems, the speech containing the words can be easily isolated and separately recognized. But speech communication used in the real world has a different pattern.

Here the word boundaries are not well separated as in the case of isolated words. Also the segmentation of the speech signal into different words is difficult and sometimes considered impossible. The frame work of IWR can not handle these complexities and we need to employ statistical techniques. These systems fall into the realm of CSR systems. It is important to note that even though IWR has a very limited recognition power it has many applications in the real world.

## 2.1 Major Approaches

The problem has been attacked using two major approaches: Template matching based systems and Stochastic Modeling based systems. Of these, the stochastic modeling approach is currently the dominant technique as it can be scaled to large vocabularies.

**Template Based Approach** This is an approach used for IWR systems. As its name suggests, the system has a prototype representation  $o_i$  of every word  $w_i \in D$ . An unknown observation sequence  $X$  will be compared to each prototype and the one with the least distance is given as the recognized word. Here the choice of the distance function is very critical to system performance. The fact that the lengths of the observation sequences are different makes the distance measure more complex. A dynamic programming based algorithm, Dynamic Time Warping(DTW)[17], is usually used to obtain the distance.

**Stochastic Approach** Let  $W \in L$  be a sentence such that  $W = w_1, w_2, w_3, \dots w_n$  where  $w_i \in D$ . The task of the speech recognizer is the following. Given an observation sequence  $X$  corresponding to sequence of words  $W$ , output the sentence  $\widehat{W}$  that, according to some criteria, best accounts for the observed sequence  $X$ . Let  $Pr(W/X)$  be the probability that the sentence  $W$  was spoken, given the observation sequence  $X$  has been observed. Let  $Pr(W)$  be the apriori probability that the sentence  $W$  is spoken. Then the recognizer should pick the sentence  $\widehat{W}$  such that,

$$Pr(\widehat{W}) = \max_W Pr(W/X) \quad (1)$$

Using Bayes' rule, the right hand side of the Equation 1 can be rewritten as,

$$Pr(\widehat{W}) = \max_W \frac{Pr(X/W)Pr(W)}{Pr(X)} \quad (2)$$

Since the maximization in Equation 1 is over a fixed  $X$ , we have from equation 1-2 a reduction of the problem to determining a sentence  $\widehat{W}$  such that,

$$\widehat{W} = \operatorname{argmax}_W Pr(X/W)Pr(W) \quad (3)$$

The equation 3 is the basis of stochastic or statistical speech recognition system. This equation leads to the modularization of the system as shown in figure 2.1. One of the modules is the feature extractor. It is also known as the Front-end subsystem of the ASR. The observation sequence  $X$  is the output of this system. We need to find tools which would provide  $Pr(W)$  and  $Pr(X/W)$ . We can see that  $Pr(W)$  is dependent on the characteristics of the language  $L$ . Similarly  $Pr(X/W)$  connects the word sequence  $W$  and the observation sequence  $X$ . These methods fall in the realm of *Language Modeling* and *Acoustic Modeling*, respectively. The following sections describe each subsystem in detail.

## Acoustic Phonetics

In a spoken language, a phoneme is a basic unit of sound that can distinguish words(that is, changing a phoneme in a word, produces another word having different meaning). In that way, it is defined as the smallest contrastive unit in the sound system of a spoken language. Phonemes are not physical sounds, but abstractions. Phonemes are regarded as a set of phones, that are regarded as a single sound, and represented by a common symbol. An allophone is one of the several phones that belong to the same phoneme. A phoneme has different acoustic realizations(allophones) based on the phonetic context. The common notation used in linguistics employs slashes(/ /) around a symbol representing a phoneme and square bracket([ ]) around a symbol representing an allophone. For example in English, words *cat* and *rat* each have three phonemes, /kæt/ and /ræt/. A pair of words that are identical, except for a single phoneme are known as a minimal pair. [p<sup>h</sup>] as in *pin* and [p] as in *spin* are two allophones of /p/.



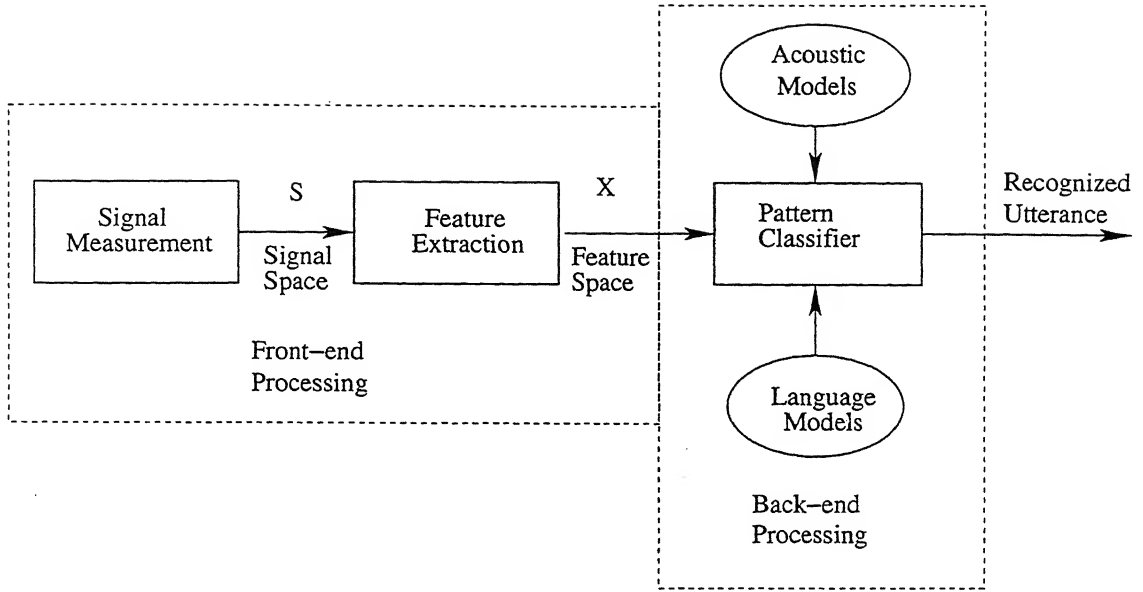


Figure 2.1: General Speech Recognition System

The acoustic representation of the phoneme can vary significantly with speaker and the context of the phoneme (place within utterance; neighboring phonemes), that is the phoneme-acoustic mapping is not one-to-one. The same phoneme displays various acoustic representation based on the speaker, the context and background noise. The articulation of one acoustic unit has its effect on the next unit to be articulated. This problem is known as *coarticulation*. To counter the effect of *coarticulation*, one needs to include context information while designing the base units. It is also a design decision as to how much context is put in the base units. Following are some of the base units.

- **Biphones**

In Biphones, one previous or next context is used to define base units giving rise to Left-context Biphone and Right-context Biphones respectively. If the phoneme set  $P$  is of size  $N$  then the Biphones would yield a base set of size  $N^2$ .

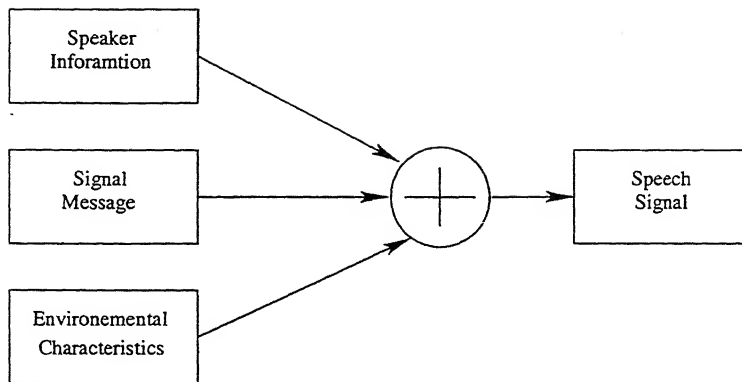


Figure 2.2: Components Contributing to Speech Signal

- **Diphones**

A diphone is defined as the last half of one phoneme  $i$  and first half of phoneme  $j$  where phoneme  $i$  and  $j$  occur consecutively. This kind of unit is used to model the transient behavior at the junction point of two phonemes.

- **Triphones**

A triphone is defined by its left and right context. That is,  $i_jk$  where  $i, j, k \in P$  represents a triphone which has a central phoneme  $j$  preceded by  $i$  and followed by  $k$ . This is the most widely used base unit set in speech recognition as it nicely captures the coarticulation effects. One of disadvantage with using triphones is the size of the set which is  $N^3$ .

There is always a trade off between the amount of context taken and recognition accuracy. Increasing the context would increase the accuracy but it also increases the base set size. It is normally seen that the triphone is an optimal solution.

## 2.2 Feature Extraction

Feature extraction is perhaps the fundamental problem in pattern recognition because it is the first step in building a recognizer. The purpose of a feature extractor is to identify, with in the data, what information is needed to perform accurate

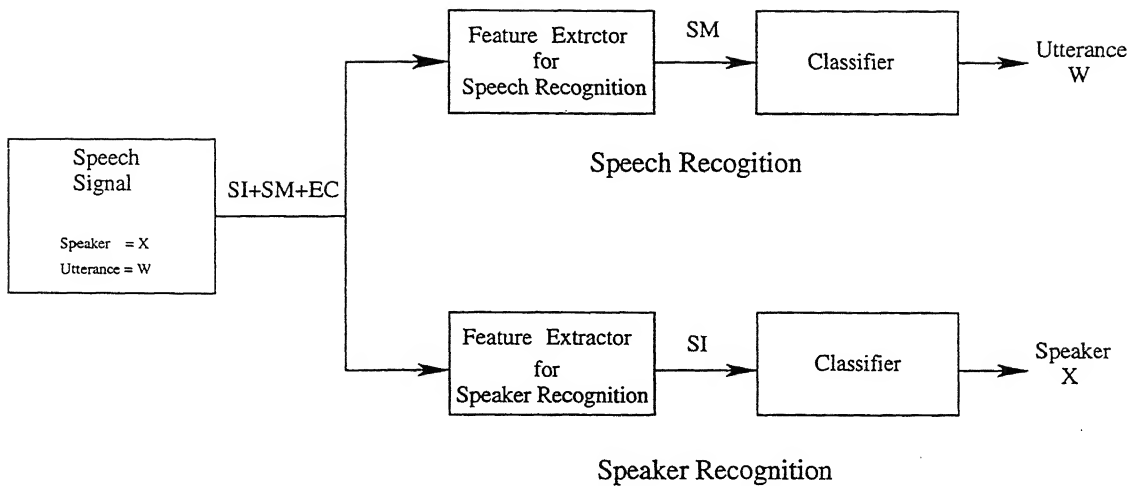


Figure 2.3: Role of Feature Extraction in Speech Recognition and Speaker Recognition

classification. The feature extraction process is expected to discard irrelevant information to the task while keeping the useful one. The following properties are required for a good feature extractor:

- Compact features to enable real time analysis
- Minimize the loss of discriminant information

A feature extractor for ASR has a specific task to perform. As shown in figure 2.2, the speech signal contains the characteristic information of the speaker(SI) and environment(EC) in addition to signal message(SM). An FE for speech recognition needs to maximally discard the SI and EC information and only allow the SM information to pass, on the other hand an FE for speaker recognition task needs to filter the SI information from the speech signal. Figure 2.3 shows an ideal FE for speech and speaker recognition task. The ability of FE for speech recognition improves depending upon how well SI and EC are filtered out:

- $SI \implies$  Speaker Independence
- $EC \implies$  Noise Robustness

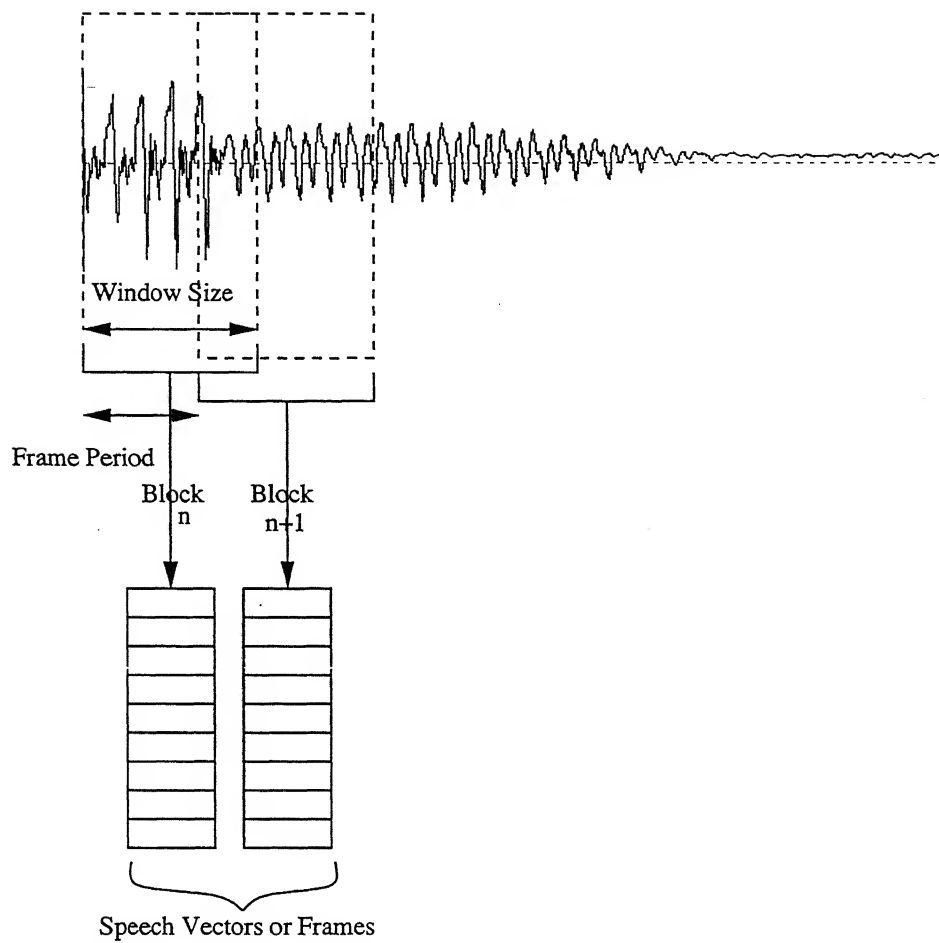


Figure 2.4: Frame based feature extraction

Mathematically, it is a transformation from the input space( $\Phi_s$ ) to feature space( $\Phi_x$ ).

$$\mathbb{F}_\theta : \Phi_s \rightarrow \Phi_x \quad (4)$$

Here equation 4 shows that  $\mathbb{F}_\theta$  is a transformation function parameterized by  $\theta$ . Hence, variation in  $\theta$  would yield different feature spaces. Let the set of such spaces be  $\chi$ . The following are some interesting properties of this set that can be investigated.

- Is there a feature space  $\Phi_x$  which separates the target class completely?
- Which is the most discriminating feature space for a discrimination measure?

In this thesis, we provide a mechanism to answer such questions. After an abstract view of a feature extractor module, let us now see specific feature extractors in use in current recognition systems. The speech signal is processed in frames with frame size ranging from 15 to 25 milliseconds and an overlap of 50%-70% between consecutive frames as shown in figure 2.2. Hence, the speech is processed on a frame-by-frame basis. The overlap between two consecutive frames is necessary in order to account for the possibility of a split of an acoustic unit. There are two major methods to extract features from each frame.

1. Linear Predictive Coefficients(LPC) - Temporal Features
2. Filter-bank based cepstrum features - Spectrum Features

In the LPC method, we try to approximate the speech segment by a linear equation. The amplitude of the signal at time  $n+1$  is determined using previous  $(k+1)$  samples already seen. These coefficients  $\alpha$  are computed by minimizing the prediction error on the entire speech segment.

$$x(n+1) = \alpha_k x(n) + \alpha_{k-1} x(n-1) + \alpha_{k-2} x(n-2) + \dots + \alpha_0 x(n-k) \quad (5)$$

The other more prevalent method is **filter-bank based cepstrum features**. This method of feature extraction would be described in detail as further optimization experiments are done on this method. Figure 2.5 shows a block diagram of a

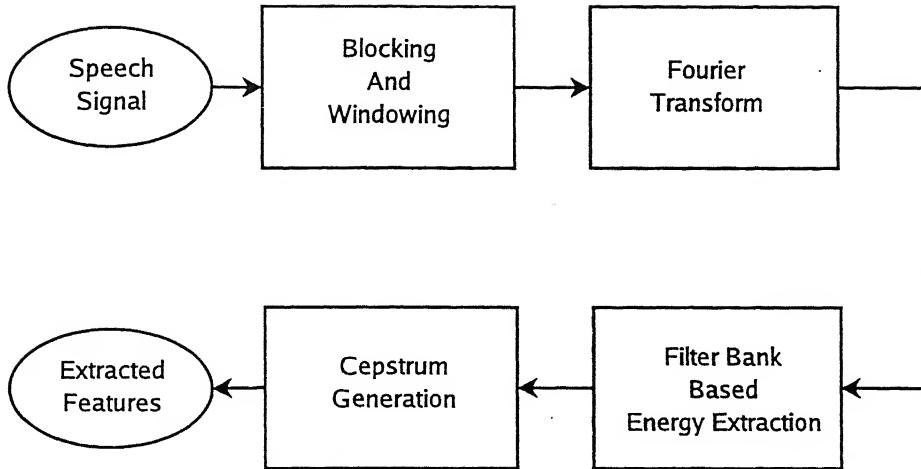


Figure 2.5: Filter-bank based feature extraction

filter-bank based feature extractor. The processing in each block is explained below. Here the speech signal is  $s(n)$ .

### 1. Blocking and Windowing

Here the speech signal  $s(n)$  is segmented into overlapping blocks and a window function is applied to obtain a windowed signal. This windowing operation is performed to reduce the discontinuities at the boundaries of the signal segment. Let the length of the segment be  $N$ .

### 2. Fourier Transform

As the filtering operation is applied in the frequency domain, the signal is transformed to frequency domain using Discrete Fourier Transformation(DFT). DFT transforms the signal from discrete time domain to discrete frequency domain. By discrete frequency domain, we mean that the frequency domain is sampled into finite number of points( $\mathbb{F}$ ). DFT is computed efficiently in practice using Fast Fourier Transform(FFT) algorithm. This algorithm requires the input signal length to be power of 2. Hence the signal is zero-padded to

make its length power of 2. Further, the squared magnitude of the spectrum vector is taken to obtain power spectrum vector  $\mathbf{x} = \{x_1, \dots, x_f, \dots, x_{\mathbb{F}}\}$ .

Any time-frequency transformation can be used in this block, but in almost all the systems only Fourier Transform is used. The other transformation that is now being applied is Wavelet Transformation.

### 3. Filter-Bank Based Energy Extraction

In this block, the energy of the signal in different frequency bands is obtained for further processing. This task is performed by a filter corresponding to a frequency band. A filter can be defined as a mechanism to pass or suppress energy contained in certain bands. We will use the implementation of a filter in the frequency domain only. The filter can have different shapes triangular, rectangular, Gaussian etc. depending upon the requirement. A filter  $F$  with any shape can be represented using a  $\mathbb{F}$  dimensional weight vector.

$$F = \{w_i | i = 1, \dots, \mathbb{F}\} \quad (6)$$

The energy of such a band can be obtained by,

$$y_j = \sum_{i=1}^{\mathbb{F}} x_i w_i \quad (7)$$

A filter-bank is a set of filters(say  $N$ ). Hence the output of this block  $\mathbf{y}$  is a energy vector of length  $N$ .

$$\mathbf{y} = (y_1, \dots, y_j, \dots, y_N) \quad (8)$$

where  $y_j$  corresponds to  $j^{th}$  filter.

### 4. Cepstrum Generation

Here a Discrete Cosine Transformation(DCT) is applied to a log-compressed energy vector producing cepstral coefficients as shown below.

$$z_k = \sum_{n=1}^N y_n \cos \frac{\pi(2n-1)(k-1)}{2N}, k = 1, \dots, N \quad (9)$$

DCT has a very good energy compaction property transforming most of the signal energy to the first few coefficients. Therefore only first few (typically 12) coefficients are taken to form the feature vector dropping the rest.

## Filter-Bank

Here we see that the filter-bank is a crucial parameter in the transformation of the signal to the feature space. The filter-banks are generally designed using the knowledge of the human auditory system. The human auditory system processes the signal in various frequency bands with linear distribution in the initial part of the frequency range and becomes non linear as we go to the end of the frequency range. To model such a system, two popular frequency scales namely Mel and Bark have been suggested. Mel scale was introduced by Davis and Mermelstein[7] in 1980 when they combined triangular filters perceptually placed with log-compressed filter output energies. The original scale contained 20 filters, first 10 linearly placed between 100 to 1000Hz, next 5 log-spaced between 1KHz and 2KHz, and 5 log-spaced between 2KHz and 4KHz. Since then other researchers tried to provide modifications to the original filter-bank out of which the scale function (equation 10) provided by Fant[10] is used in most MFCC systems.

$$\hat{f} = 2595 \log_{10} \left( 1 + \frac{f}{700} \right) \quad (10)$$

On this scale, filters are equally spaced with each bandwidth determined by the frequency range  $(\hat{f}_{min}, \hat{f}_{max})$  and number of filters  $N$ . Thus:

$$\Delta \hat{f} = \frac{\hat{f}_{max} - \hat{f}_{min}}{N + 1}$$

$$\hat{f}_{ci} = \hat{f}_{min} + i \Delta \hat{f} \quad i = 1, \dots, N \quad (11)$$

where  $\hat{f}_{ci}$  is the central frequency of the  $i^{th}$  filter. In a triangular filter-bank, the left( $\hat{f}_{li}$ ) and right( $\hat{f}_{ri}$ ) frequency is the center frequency of the previous and next



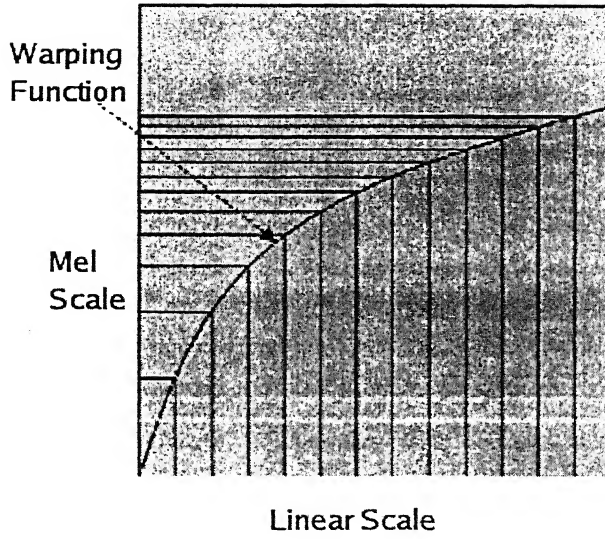


Figure 2.6: Mel frequency scale approximation proposed by Fant

filter respectively. They are defined as shown in the following equations.

$$\hat{f}_{li} = \begin{cases} \hat{f}_{min} & i = 1 \\ \hat{f}_{ci-1} & 1 < i \leq N \end{cases} \quad (12)$$

$$\hat{f}_{ri} = \begin{cases} \hat{f}_{ci+1} & 1 \leq i < N \\ \hat{f}_{max} & i = N \end{cases} \quad (13)$$

## 2.3 Classification

### 2.3.1 Acoustic Modeling

In this subsystem, the connection between the acoustic information and phonetics is established. The connection can be established either at word or at phoneme level gives rise to word based system and phoneme based system, respectively. In both cases a speech unit  $p$  is mapped to its acoustic counterpart using temporal models as speech is a temporal signal. There are many models for this purpose like,

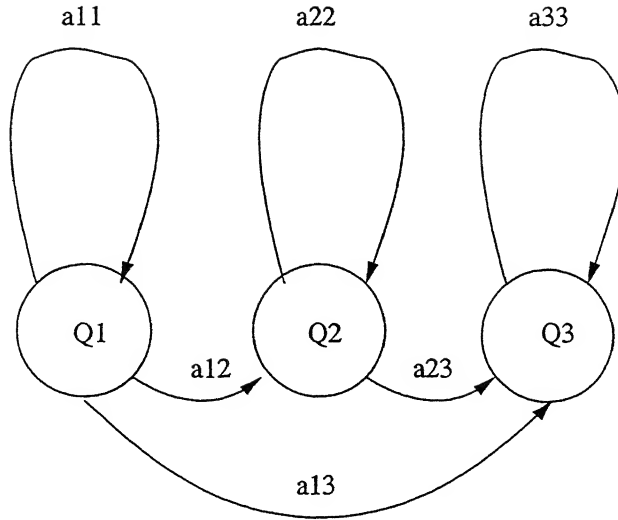


Figure 2.7: A three state left-to-right HMM

- Hidden Markov Model(HMM)
- Artificial Neural Network(ANN)
- Dynamic Bayesian Network(DBN)

ANN is a general pattern recognition model which found its use in ASR in the early years. Rabiner[18], in 1991, first suggested the HMM approach leading to substantial performance improvement. Current major ASR systems use HMM for acoustic modeling. Since then, researchers have tried to optimize this model for memory and computation requirements. In the current state, it seems that HMM has given the best it could and now we need to find other models to go ahead in this domain. This leads to consideration of other models in which Dynamic Bayesian Network seems a promising direction[27]. As our experiments and verification has been done on a HMM based system, we have a detailed look at HMM.

**Hidden Markov Model** The basic idea of HMM is that the observation sequence  $X$  is generated by a system which exists in one of a finite number of states. At each time step, the system makes a transition from the current state to the next

state while emitting an observable quantity according to a state specific probability distribution. Precisely speaking, a hidden markov model  $M$  is a four tuple consisting of the following.

1. A set of possible states  $Q$
2. A transition matrix  $A$  where  $a_{ij}$  is the probability of making a transition from state  $q_i$  to state  $q_j$
3. A state conditioned probability distribution over observations, that is a specification for  $Pr(x/q_i)$  for any observation  $x$  parameterized as  $B$
4. An initial state probability  $\pi$

The observation sequence modeled by the HMM may be discrete or continuous in nature, but the state space can only be discrete. This model has a very well studied mathematical background of Markov Processes. Basically, an HMM is a first order Markov process with state emitting observations. Figure 2.7 shows a three state HMM with all the transitions either going forward or loop back to itself. This kind of HMM is called a left-to-right HMM which can naturally model the speech signal. There are various kinds of HMM topologies which impose restrictions on the transitions allowed between states.

There are three basic problems related to HMM all of them important to speech recognition. The algorithms to solve these problems are the important places for improvements as they are at the core of HMM based speech recognition.

### 1. The Evaluation Problem

Given an HMM  $M$  and an observation sequence  $X = x_1, x_2, x_3, \dots, x_T$ , what is the probability that the observations are generated by the model,  $Pr(X/M)$ ?

### 2. The Decoding Problem

Given an HMM  $M$  and an observation sequence  $X = x_1, x_2, x_3, \dots, x_T$ , what is the most likely state sequence that produced the observations?

### 3. The Learning Problem

Given an HMM  $M$  and an observation sequence  $X = x_1, x_2, x_3, \dots, x_T$ , how should we adjust  $(A, B, \pi)$

The problems are addressed by the following algorithms: **Forward Algorithm**, **Viterbi Algorithm** and **Baum-Welch Algorithm** respectively. We will not go into the details of these algorithms. A detailed discussion of the same is available here[18].

To see, how it can be applied to ASR, we will study an isolated, whole-word isolated word recognizer. In this system, there is a HMM  $M_i$  for each word in the dictionary  $D$ . HMM  $M_i$  is trained with the speech samples of word  $w_i$  using the Baum-Welch algorithm. This completes the training part of the ASR. At the time of testing, the unknown observation sequence  $X$  is scored against each of the models using the forward algorithm and the word corresponding to the highest scoring model is given as a recognized word.

#### 2.3.2 Language Modeling

The goal of language modeling is to produce accurate value of  $Pr(W)$ . A language model contains the structural constraints available in the language to generate the probabilities. Intuitively speaking, it determines the probability of a word occurring after a word sequence. It is easy to see that each language has its own constraints for validity. The method and complexity of modeling language would vary with the speech application. For example, a simple speech enabled call dialing system which would have a very limited vocabulary and constrained input will have a simple language model. On the other hand, the task of transcribing broadcast news data would require a large vocabulary of the order of thousands with sentence structure that is much less constrained. This leads to mainly two approaches for language modeling as described below. The appropriateness of the approach is problem specific. Generally, small vocabulary constrained tasks like phone dialing can be modeled by *grammar based approach* where as large applications like broadcast news transcription require *stochastic approach*.

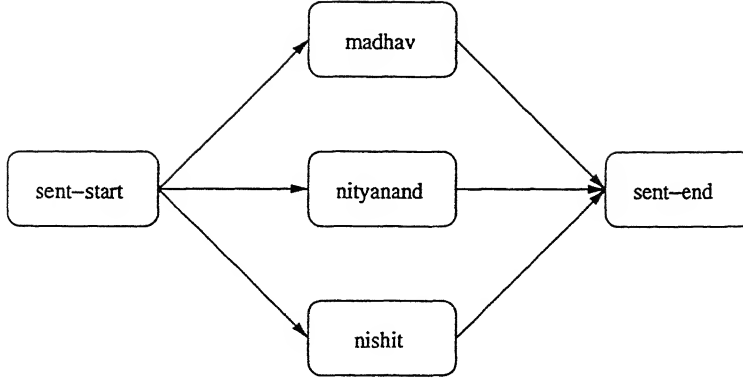


Figure 2.8: Search graph generated by grammar

### Grammar Based Models

In this approach, the structure of the language is defined in term of grammar rules. Generally, this grammar is a context free grammar. Hence, from the grammar, we can generate a search graph where the nodes will be the words in the dictionary. For example, the search graph generated by the grammar shown below is shown in the figure 2.8. Hence, a path from <sent-start> to <sent-end> would correspond to a correct hypothesis.

SENTENCE = <sent-start> NAME <sent-end>

NAME = madhav | nishit | nityanand

### Statistical Language Models

Here, the  $Pr(W)$  is determined using the chain rule as shown in Equation 14.

$$Pr(W) = \prod_{i=1}^n Pr(w_i/w_{i-1}, w_{i-2}, \dots, w_0) \quad (14)$$

Here, the probability of a word  $w_i$  occurring in a sentence is dependent on all the past words. It is very apparent that this would require a large number of probabilities to be estimated. The number of probabilities to be estimated is exponential in terms of vocabulary size. Therefore an approximation to this method is required.

This can be achieved by truncating the past dependencies which would give rise to equation(15).

$$Pr(W) = \prod_{i=1}^n Pr(w_i/w_{i-1}, w_{i-2}, \dots, w_{i-N+1}) \quad (15)$$

Commonly used values for N are trigram  $N = 3$ , bigram  $N = 2$  and unigram

$N = 1$ . Such language models need huge data from the corresponding language

to estimate the probabilities. It is easy to see that as we increase the number N, the data requirement increases exponentially. A trivial method to get these probabilities from the data is to obtain word frequencies. As the amount of data increases the approximation in estimating the probabilities would tend to the actual values. The probability information for a trigram model can be estimated using the equation below.

$$Pr(w_i/w_{i-2}, w_{i-1}) = f(w_i/w_{i-2}, w_{i-1}) = C(w_{i-2}, w_{i-1}, w_i)/C(w_{i-2}, w_{i-1}) \quad (16)$$

where  $C(\cdot)$  represents number of occurrences of the corresponding word sequence. There is a major problem in using raw frequencies as probabilities. Any word combination, which is not present in the training corpus, would be assigned probability value zero. Therefore it is necessary to smooth trigram frequencies using the following equation.

$$Pr(w_i|w_{i-2}, w_{i-1}) = \lambda_3 * f(w_i|w_{i-2}, w_{i-1}) + \lambda_2 * f(w_i|w_{i-1}) + \lambda_1 * f(w_i) \quad (17)$$

where nonnegative weights  $\lambda_1 + \lambda_2 + \lambda_3 = 1$ . The other approach used to alleviate this problem is clustering the words into different classes. Then the probability estimation is done on different classes instead of words. The actual word probabilities are then derived from the class probabilities.

### 2.3.3 Decoding Algorithm

As we can see from equation(3), the problem of finding the correct sentence is an optimization problem, once  $Pr(W)$  and  $Pr(X/W)$  are known. The behavior of

the ASR system heavily depends on the decoding algorithm as that would mainly determine the response time of the system. The language model would induce a search graph containing nodes as words and links between them representing the probabilistic relationship between them. Here we note that the computation of  $Pr(W)$  is independent of the observation sequence  $X$ , where as  $Pr(X/W)$  has to be computed at run time using the model parameters. Decoding problem is to find the best path in the search graph. Therefore the algorithm involves has to maintain a set of active paths(paths which can possibly reach the sentence end) and keep expanding them until the end of speech is reached. After that, the hypothesis corresponding to the path having the highest probability is given as the recognized hypothesis. A path would contain a sequence of states of the graph. One way of search is to keep paths of the same length, that is each path is scored against last speech frame obtained. This is called *Time Synchronous* decoding. Most widely used Time Synchronous decoding is the Viterbi algorithm. The other way is to keep expanding the path with highest probability. When the probability of the best path drops, we need to select other path in the list and apply previous frames which were read and buffered to reach current frame. This is called *Time Asynchronous* decoding. Stack based decoding algorithm are modification of  $A^*$ -algorithm which falls into Time Asynchronous category.

This chapter provided a brief overview of the speech recognition process and tools used for that purpose with emphasis on **Feature Extraction**. An exhaustive survey of the speech recognition method and their evolution has been provided in [13, 19]. In the next chapter, we introduce the basic concepts used in this thesis and discuss previous work.

# Chapter 3

## Previous Work and Preliminaries

### 3.1 Optimization of Features for pattern recognition

Optimization of feature extraction systems to obtain improvements in the classification accuracy is a well-studied field in pattern recognition. Traditionally, the features for a pattern recognition task are defined using expert knowledge. This approach of designing a feature extractor has been successful for some pattern recognition tasks. However, as the problem becomes more and more complex, the design of features becomes more difficult because of the increase in the dimensionality and classification complexity. Let us take an example task of designing a feature extractor for face recognition that is classifying a type of flower into the classes, let's say A and B. The trivial features that come up are color, length of the petal, blooming period, and number of petals. A simple analysis of the spread of the samples of two classes with respect to different features would yield the feature set required. The set of features for which the classes show up well clustered and non-overlapping can become an ideal feature set. From this recognition task, let us go to a task of recognizing human face. Here also there would be many candidate features present but the dimensionality of such task makes it intractable for manual analysis.



Speech recognition is one such problem where knowledge of human auditory system has been utilized while designing its feature set. Processing of speech through various nonlinear frequency bands by auditory system gave rise to nonlinear frequency band energy based feature set with bark and mel scale. Various researchers like Fant[10] and Slaney[22] have suggested modifications of these scales reporting improvements in the recognition accuracy. These features enabled speech recognizers transcribe speech with high accuracy in constrained conditions. But these system degrade drastically with change in speaker characteristics and environmental conditions. The most probable reason for such a behavior can be drastic variation in the features extracted in the changed conditions. This situation demanded for features which are robust to these changes.

This problem motivates the use of labeled data to obtain optimal features. There are several ways to use the labeled data in pattern recognition in general and speech recognition in particular. We briefly discuss some approaches here.

### Artificial Neural Network(ANN)

ANNs have proved their ability to solve difficult classification problems. Their ability to adapt with a robust training algorithm makes it one of the obvious methods to obtain data dependency in pattern recognition. The ANN/HMM hybrid approach for speech recognition is an example where ANN is being used as a feature extractor. In this approach, the training task comprises of learning the HMM parameters through Baum-Welch algorithm and simultaneously updating the neural network weights to generate better features. A state of the art survey for this approach is provided by Edmondo Trentin et. al.[24]

### Transformation of Feature Vectors

In this approach, the features are first extracted using the conventional feature extraction system and then a transformation is applied to obtain enhanced features. The transformation  $\rho$  is generally from high dimensional space  $\mathbb{R}^n$  to a lower dimensional space  $\mathbb{R}^p$  maintaining the discriminative information reducing redundancy.

$$\hat{O}(\tau) = \rho O(\tau) \quad (1)$$

In above equation, a  $n$ -dimensional feature vector  $O(\tau)$  is transformed into a  $p$ -dimensional vector  $\hat{O}(\tau)$  using an  $p \times n$  dimensional transformation matrix  $\rho$ . One basic method for linear transformation is to apply Principal Component Analysis (PCA)[8] which is an unsupervised method as it does not require class information. This method finds an orthogonal coordinate system in which the axes are ordered according to decreasing variance. The first few coordinates take up most of the variance present in the data. Hence projection of the original data on the newly generated truncated coordinate system yields dimensionality reduction. Linear Discriminant analysis is one standard method in pattern recognition which takes the class information into account. Here the transformation  $\rho$  can be found using any class scatter metric as an optimization criteria. Xunying Liu[14] has used linear transformation based optimization using modified LDA.

### Parameterization and Optimization of Feature Extractor

In this approach, the feature extraction system is parameterized and then optimization is performed to obtain the parameters. A formalism for such an approach was first provided by Biem et al.[2][3] when he used this approach to obtain features for the speech recognition task. This paradigm assumes that the feature extractor and recognizer are both parameterized as  $\theta$  and  $\rho$  respectively. During the training phase, both the parameters are learned together. This method is applied to small speech recognition tasks with a filter-bank based feature extractor treating the filter-bank parameters as the parameters to optimize. The aim of the experiment is to obtain a better filter-bank which can then be used for a larger recognition task. The method used for optimization is generally gradient based using an objective function which tends to simulate the recognition system. The method is mostly applied to filter-bank based feature extractors because of its widespread use in speech recognition systems.

Biem[3] employed vowel classification as a recognition task. The center fragment of each vowel was used to generate 256 FFT-based spectral coefficients through a hamming window. A neural network was used as a classification mechanism. The classification output obtained from the neural network was used to compute the

objective function used in the gradient descent search.

Hermansky et. al.[15] use LDA to optimize the filter-bank. Here the within class and between class covariance is used as a class scatter metric. They have reported improvement in the recognition accuracy through optimized features. They also provide an analysis of the shape of the filters obtained which could lead to an understanding of the type of spectral changes that carry phonetic information.

The bandwidths of the filters of the filter-bank also have an effect on the discriminative ability of the features which can be conclude from Mark[21]'s work. In his work, the bandwidth of the mel filter-bank is modified giving out new features called HFCC. Here also the within class and between class measure is used using phonemes as classes. One of the problems here was the truncation of the phoneme at the boundaries to make the length of all the phonemes equal. This loss of length information can lead to improper optimization as the length is also a critical parameter used by the human perception system. Therefore a mechanism to generate constant length representation from variable length data is required. Varying the filter parameters also has an effect on the robustness of the system to noise. This can be concluded from the work of Torre[6].

The previous work reviewed above suggests that more exploration in filter-bank based feature extraction can lead to improvements in speech recognition systems and help us create a system which can work in an unconstrained environment. In our work, we try to optimize the filter-bank using a different optimization mechanism and objective function. All the previous work uses gradient based optimization which put constrains the definition of the objective function. We use an evolutionary approach, *Genetic Algorithms*, which does not require the objective function to be differentiable. The objective function is defined using criteria similar to within class and between class variances with phonemes used as classes. But no truncation is performed. The features obtained for phoneme samples are transformed into constant length features using time series formulation methods.

## 3.2 Genetic Algorithm as an Optimization Method

In this section, we briefly review the evolutionary technique on genetic algorithms (GA). GA is a search and optimization method which mimics the natural and chromosomal processing in natural genetics. These algorithms try to mimic the natural process of evolution to solve an optimization problem. In some other optimization problems, the search space is astronomically large prohibiting the brute force approach to find an optimal solution. Many of the problems, involving optimizing the feature extractor parameters having real values, lead to search spaces that are infinite. The classical optimization methods like gradient based optimization have been used to solve such problems. One major drawback of classical methods is the requirement of a differentiable objective function. Some problems have a non-differentiable objective function that makes classical methods inapplicable.

GA simulates the survival of the fittest among individuals over consecutive generation for solving a problem. Each generation consists of a population of character strings that are analogous to the chromosome. Each individual represents a point in a search space and a possible solution. The individuals in the population are then made to go through a process of evolution.

GA is based on an analogy with the genetic evolution. The basic principles are:

- Individuals in a population compete for resources and mates
- Those individuals most successful in each ‘competition’ will produce more offspring than those individuals that perform poorly
- Genes from ‘good’ individuals propagate throughout the population so that two good parents will sometimes produce offspring that are better than either parent
- Thus each successive generation will become more suited to their environment

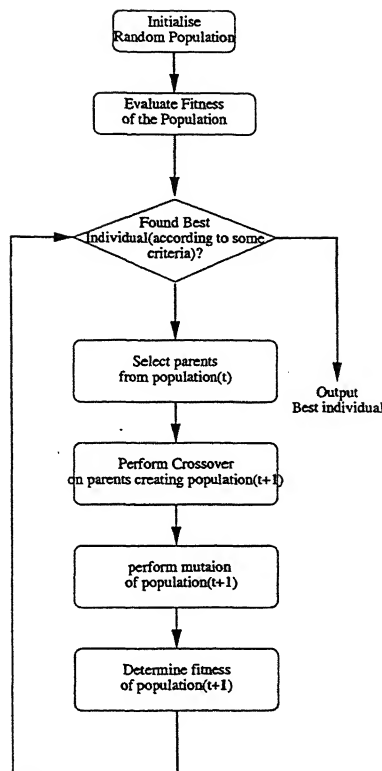


Figure 3.1: Genetic Algorithm

## Search Space

A population of individuals is maintained within a search space for a GA, each representing a possible solution to a given problem. Each individual is coded as a finite length vector of components, or variables, in terms of some alphabet. To continue the genetic analogy these individuals are likened to chromosomes and the variables are analogous to genes. Thus a chromosome (solution) is composed of several genes (variables). A fitness score is assigned to each solution representing the abilities of an individual to 'compete'. The individual with the optimal (or generally near optimal) fitness score is sought. The GA aims to use selective 'breeding' of the solutions to produce 'offspring' better than the parents by combining information from the chromosomes.

The GA maintains a population of  $n$  chromosomes (solutions) with associated fitness values. Parents are selected to mate, on the basis of their fitness, producing offspring via a reproductive plan. This mating plan consists of the sequence of genetic operators to be applied to the current population to generate the next generation. Consequently highly fit solutions are given more opportunities to reproduce, so that offspring inherit characteristics from each parent. As parents mate and produce offspring, room must be made for the new arrivals since the population is kept at a static size. Individuals, in the population, die and are replaced by the new ones, eventually creating a new generation once all mating opportunities in the old population have been exhausted. In this way it is hoped that over successive generations better solutions will thrive while the least fit solutions die out.

New generations of solutions are produced containing, on average, better genes than a typical solution in a previous generation. Each successive generation will contain better 'partial solutions' than previous generations. Eventually, once the population has converged and is not producing offspring noticeably different from those in the previous generations, the algorithm itself is said to have converged to a set of solutions to the problem at hand. This entire process is shown in figure 3.2.

### 3.2.1 Genetic Operators

Here, we describe the function of each operator.

#### Selection

This operator is used to select the parent elements from the current population pool. The key property of this operator is the ability to give more preference to better individuals to allow them to pass better genes to the next generation. Here the goodness of each individual is dependent on the fitness assigned by the objective function. A trivial selection operator would select the best  $N$  individuals for reproduction. But this may not be the best strategy as weak individuals might also possess some good genes that, if overlooked, would never come into the best solution set leading to a sub-optimal solution.

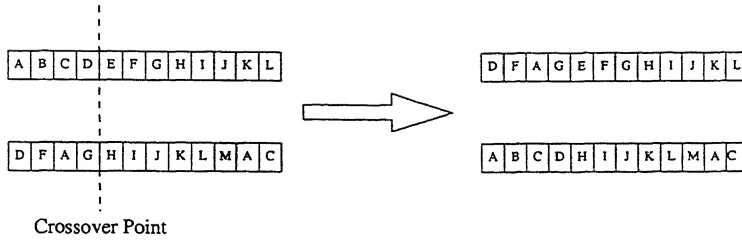


Figure 3.2: Crossover operation

## Crossover

This is an operator which induces diversity in the population. Two individuals from the current population( $t$ ) are chosen using the selection operator and a random point of crossover is generated. Now the portion of the chromosome up to the randomly generated point is swapped between two individuals giving rise to two new individuals. These newly generated individuals are put into the next generation pool as shown in figure 3.2.1. The crossover between parent individuals is performed with probability  $p_c$ , which is also called crossover rate. That is, in  $100p_c\%$  of the crossover operation, new offsprings are put in the next generation. Rest of the time, parent individuals are directly passed with any change. Intuitively, we are recombining the portions of chromosomes of good individuals which are likely to produce even better individuals.

## Mutation

The function of this operator is to randomly change part of the chromosome with low probability(mutation rate- $p_m$ ). The purpose of this operator is to maintain the diversity of the population and to avoid premature convergence. This operator alone would induce a random walk through the search space. The implementation of this operator is heavily dependent on the kind of chromosome being evolved. For example, this operator may simply flip some bits of a binary coded GA, while for chromosomes having real elements the task is to perturb some of the values using random numbers.

There are two types of GA being used. They are different in terms of how the new population is generated from the previous one. One option is to generate new offspring and put them into the next generation without considering the parent generation which is done in *Simple GA*. The other way of generating the next population is to merge the parent and child population and then select the required number of individuals from the entire pool which is done in *SteadyState GA*. The advantage of this method is that the 'good' individuals from the parent generation which failed to have an effect in the current process can remain in the pool for the next generation. In our implementation, we use *SteadyState GA*.



# Chapter 4

## Design of GA Based Feature Extraction Optimization System

In this chapter, we will describe the design of a genetic algorithm based feature optimization system. The design of such a framework can be broken into two major aspects.

1. Optimization Method
2. Objective Function

Any optimization method searches for the best solution in the search space induced by the parameter set. In the feature extraction process, the feature extraction method would define a search space. A filter-bank based feature extractor would induce a set of feature spaces each corresponding to a specific filter-bank. As the filter-bank contains real parameters, the set is infinite.

As described in the previous chapter, the three most important aspects of using genetic algorithms are:

1. Definition of the Genetic Representation
2. Definition of the Genetic Operators
3. Definition of the Objective Function

Defining the above three aspects completely define the optimization method. Before each aspect is discussed in detail, an abstract classification task is defined.

## Classification Task

A recognition task consists of  $M$  classes  $\{\Omega^{(1)}, \dots, \Omega^{(i)}, \dots, \Omega^{(M)}\}$ .  $N_i$  represents number of samples present for class  $\Omega^{(i)}$ . Let  $S_{ij}$  be the  $j^{th}$  sample of  $i^{th}$  class. The task is to classify an unknown sample into one the  $M$  classes. The class definition can be any one of phoneme, syllable or word. The feature set optimization process searches for the most discriminative feature set with respect to these classes ( $\Omega_i$ ) based the samples provided.

### 4.1 Genetic Representation

In filter-bank based feature extraction, the filter bank is a set of filters extracting the energy of the signal in the corresponding frequency band. Therefore, it can be said that the induced feature space  $\Phi_x$  is highly dependent on the set of the filters. A chromosome is defined as a sequence of such triangular filters. A triangular filter is represented using three frequencies:

- Left frequency -  $\alpha$
- Center Frequency -  $\beta$
- Right Frequency -  $\gamma$ .

As we have noticed earlier that a filter-bank is applied on a discrete frequency domain, we can think of the continuous frequency domain being partitioned into finite number number of bins. Hence the edge-frequencies of the filters are specified in terms of bin number instead of absolute frequency. For example, if the frequency domain is represented by 512 points, (20,45,50) is a valid tuple representing a filter.

A filter bank is a sequence of such filters. Let us say the number of filters is  $N$ . Hence, in our case, the parameter set  $FB = \{F_i | i = 1, \dots, N\}$  where  $F_i$  is a 3-tuple  $(\alpha_i, \beta_i, \gamma_i)$ . A filter would represent an element of a chromosome.

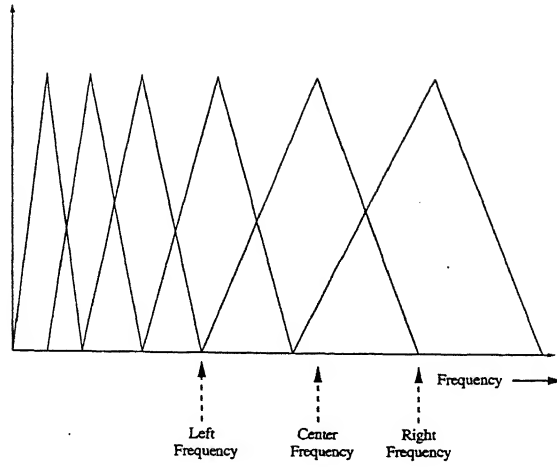


Figure 4.1: A filter bank

## 4.2 Genetic Operators

### 4.2.1 Initialization

The initialization of the population can be done in different ways. One way is to initialize all the individuals of the population randomly placing the triangular filters on the frequency axis. The other way of doing the same is to place filter-banks in the region near to the known filter-banks like Mel/Bark scale. The idea is to find the optimal filter bank which could be close to Mel/Bark scale in solution space or it might be some other filter bank. Hence, we will be initializing the filter banks using randomly perturbed Mel/Bark scale in our optimization experiments. The figure 4.1 shows a sample filter-bank generated from a Mel filter bank.

To initialize the individuals of a population, first a mel-scale based filter-bank( $FB_{mel}$ ) with  $N$  filters is generated as described in Section 2.2. Then this filter-bank is perturbed as follows:

1. randomly choose number of filters to be modified
2. randomly choose the filters to be modified

3. change the edge-frequencies of each filter selected in small neighborhood

In the last step, the filter edge-frequencies( $\alpha$ ) is changed to a random number in  $[\alpha - md, \alpha + md]$  where  $md$  is the maximum possible deviation. The order of the frequency edges(left edge < center edge < right edge) must be maintained in order to have a well formed filter-bank. Therefore, after the perturbation of a filter, this property is checked and if it is not satisfied, perturbation is performed again.

#### 4.2.2 Mutation

The mutation of a filter bank can be done by varying the filter frequencies randomly in a small neighborhood. This operator is implemented in the same way as the perturbation of a filter-bank is described in the Initialization operation.

#### 4.2.3 Crossover

The definition of this operator is not very different in this case. A random integer  $r \in [1, N]$  is generated to obtain a crossover point. At this point, the filters of the two filter-banks are swapped to generate offspring. Let  $FB_1$  and  $FB_2$  be two individuals selected for mating. They are defined as follows,

$$FB_1 = \{F_i^1 | i = 1, \dots, N\}$$

$$FB_2 = \{F_i^2 | i = 1, \dots, N\}$$

After applying the operator with randomly selected crossover point  $r$ , offsprings generated would be,

$$FB_1^{child} = \{F_1^1, \dots, F_r^1, F_{r+1}^2, \dots, F_N^2\}$$

$$FB_2^{child} = \{F_1^2, \dots, F_r^2, F_{r+1}^1, \dots, F_N^1\}$$

It is interesting to note that the filter-bank based optimization problem has a very natural genetic representation and simple implementation for the crossover operator. The objective function is the most important aspect which is described next.

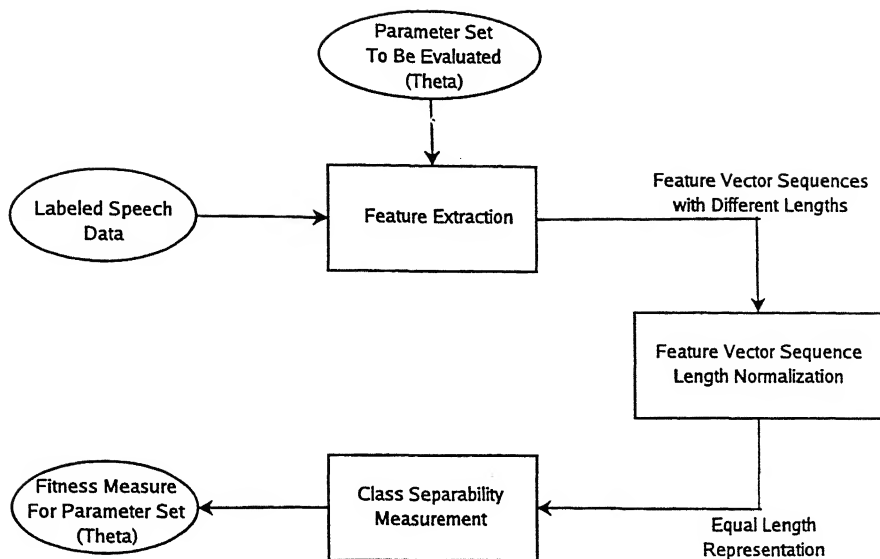


Figure 4.2: Objective Function Evaluation

## 4.3 Objective Function

This function has to check the *goodness* of a given filter-bank. A trivial way of doing this is to use such a filter bank in a feature extraction module of a complete speech recognition system and perform the training and testing task to obtain the final recognition accuracy. This would precisely indicate the *goodness*. The problem of doing this kind of evaluation lies in the computational requirements. The training task for even a small vocabulary system takes huge computational resources and takes considerable time. In the Genetic Algorithm paradigm, there would be thousands of such individuals to be evaluated in hundreds of generation. This makes such an evaluation infeasible to implement.

To evaluate the *goodness*, researchers have used alternate methods which are less computation intensive. Let us check the properties of such a function. Let  $O(\cdot)$  be the objective function and  $Acc(\cdot)$  is the accuracy of a particular recognition experiment for a feature extractor parameter set  $\theta$ . The ideal objective function

should behave according to the following relation.

$$O(\theta_1) \geq O(\theta_2) \iff Acc(\theta_1) \geq Acc(\theta_2) \quad (1)$$

The standard method to define such an objective function is *class separability* criteria. Various quantitative measure for class separability are *Bayes risk*, *variational distances*, *scatter matrix based measures*, *Bhattacharya distance* and *divergence rate*[12]. Bayes risk is the best measure of separability of distributions and for any feature space it gives the minimum amount of attainable risk. Theoretically, Bayes error is the optimum measure of feature effectiveness, but its computational complexity restricts its use for measurement purpose. The computational requirement of a method is the most important aspect considered for selection purpose as the method is to be executed by a genetic algorithm for many generations. The elegant and yet simple way of formulating a criteria for class separability is based on within-class and between-class scatter matrices which is widely used in discriminant analysis in statistics[8]. This analysis is performed using the mean and covariance of the class clusters. One disadvantage of such a measure is the fact that these criteria do not have a direct relationship to the probability of error for the Bayes classifier. The other criteria Bhattacharya measure is derived from the Chernoff Bound(upper bound of Bayes error)[12]. Also the computational requirement is equivalent to the scatter matrix based measure. Hence, in our implementation, Bhattacharya Distance has been used. This measure also uses mean and covariance of class clusters.

To obtain the means and covariances of the samples belonging to a single class and to perform the between class analysis, the representation of the samples should be in a space where each sample is a point. In case of speech samples(phonemes/syllables/words), the length of signal in signal space is varying with speaker, speaker condition and context. Therefore the frame based feature sequence extracted from the signal also has variation in length. If we analyze the samples in signal space for their length, we can find a range  $[\tau_{min}, \tau_{max}]$  for their utterance lengths. There are two options to handle this problem.

### 1. *Truncation in Signal Space*( $\Phi_s$ )

This is the standard approach applied in related work. The sample is truncated at the boundaries from the center to make its length constant across all samples. The short coming of the approach is that there will be loss of critical information that may be crucial in a recognition experiment.

### 2. *Length Normalization in Feature Space*( $\Phi_x$ )

Here, the features are obtained from different length samples. This will produce the feature sequences with different lengths. At this point, normalization is applied which transforms the samples into a constant length representation.

We use this approach in our implementation.

The procedure can be divided into three major operations as shown in the figure 4.2. Each processing block will be described in the following subsections.

## 4.3.1 Feature Extraction

This processing block takes the labeled speech data and applies the feature extractor using the parameter set provided. The parameter set provided is generated by the genetic algorithm. The block is represented with a sample  $s(n)$  of each class  $C_i$  one at a time. Here we use the Frame-Based Filter Bank Feature Extractor explained in chapter 2. Therefore the speech sample segment  $s(n)$  would result in  $\tau$  frames. Each frame would yield a corresponding feature vector of length  $\mathbb{F}$ . The feature vector sequence is described by  $X = \{x_t\}_1^\tau$ .

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,\tau} \\ x_{2,1} & x_{2,2} & \dots & x_{2,\tau} \\ \vdots & \vdots & \ddots & \vdots \\ x_{\mathbb{F},1} & x_{\mathbb{F},2} & \dots & x_{\mathbb{F},\tau} \end{bmatrix}$$

The sequence can also be represented by collecting the  $f^{th}$  vector element through time  $t = 1$  to  $t = \tau$ , that is  $w_f = \{x_{f,1}, \dots, x_{f,t}, \dots, x_{f,\tau}\}$ . Therefore the feature vector would look like,

$$X = \begin{bmatrix} w_1 \\ \vdots \\ w_f \\ \vdots \\ w_F \end{bmatrix}$$

where  $w_f = \{w_{f,1}, \dots, w_{f,t}, \dots, w_{f,\tau}\}$ .

### 4.3.2 Feature Vector Sequence Length Normalization

The length normalization is a crucial operation which is required to maintain the properties of the signal while transposing the signal to a constant length representation. This problem is classically known as the *Time Series Normalization Problem*. Let us say  $\Psi$  is the function that maps the sample from the feature space to the constant length representation.

$$Y = \Psi(X) \tag{2}$$

where  $Y$  is a normalized representation of the given sample.

$$Y = \Psi(X) \tag{3}$$

$$= \Psi \begin{pmatrix} w_1 \\ \vdots \\ w_f \\ \vdots \\ w_F \end{pmatrix} \tag{4}$$

$$= \begin{pmatrix} \psi(w_1) \\ \vdots \\ \psi(w_f) \\ \vdots \\ \psi(w_F) \end{pmatrix} \tag{5}$$

The task of the function  $\psi$  is to represent the given sequence into constant length representation retaining the property of the input. Spectral methods are used for



this purpose. DFT is generally used to transform the signal into frequency domain. For large time series data, the dimensionality reduction is done through this method as described in [9, 1].

Here, the dimensionality reduction is used to search the time series databases. Recently, DWT based methods have been used for the same purpose because of their better performance[16]. In all these methods, the transformation is from higher dimensional(but constant) to lower dimensional space. The time series in the input space have equal lengths and hence they are transformed into other space. In our work, we have used another spectral method, DCT, as a transformation method because of its energy compaction property. We also tried polynomial regression as a method for time series representation, but after some experimentation it was found that the DCT method outperforms polynomial regression.

## ■ Discrete Cosine Transform (DCT) - $\psi_{DCT}$

We briefly introduced this transformation in Chapter 2. Some of the nice properties of this transformation make it suitable for time series normalization. This transform is known for its ability to compact the energy of the signal into the first few coefficients of the transform. The image compression standard JPEG[26] is primarily based on this property. We use the compression ability of DCT for time series normalization. The DCT transform equation is given below:

$$z_k = \sum_{n=1}^N y_n \cos \frac{\pi(2n-1)(k-1)}{2N}, k = 1, \dots, N \quad (6)$$

Basically, it provides a one-to-one mapping between the input sequence and the transformed sequence having the same length. The inverse transformation IDCT can be used to retrieve the original signal without information loss. The compression is achieved by dropping some of the coefficients of the transformed vector. Generally these coefficients contain very less energy which accounts for equally less information in the signal. Figure 4.3 shows the original signal of length 1258 at the top which is an aperiodic signal. A DCT is applied to this signal and only the first 200 coefficients(approximately 1/6th) are kept making all others zero. The signal at the bottom is the reconstructed one.

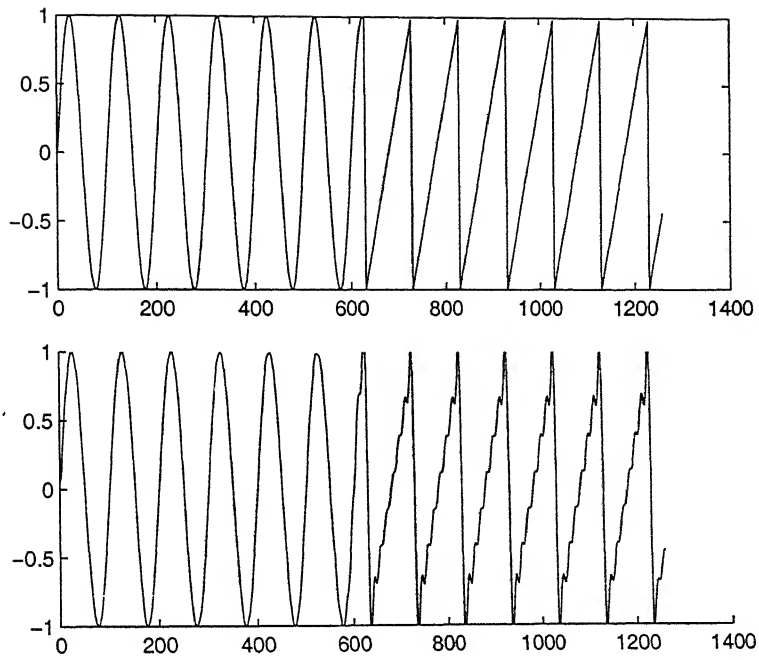


Figure 4.3: Top: Original Signal Bottom: Reconstructed Signal

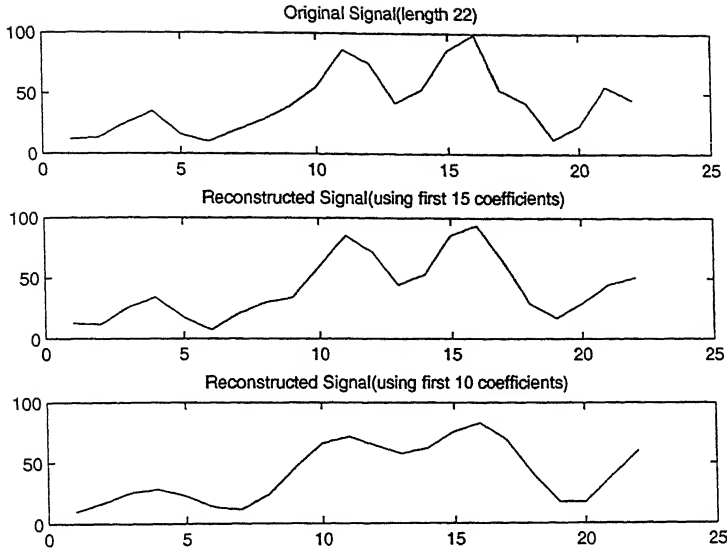


Figure 4.4: Original and Reconstructed Signals

Figure 4.4 shows the sequence that is very much similar to the one encountered in our normalization problem. A general range of sequence length is 15-25 frames. The second and third signals are reconstructed from the first 15 and 10 DCT coefficients respectively. In our experiments, a normalization length of 15 is used.

### Decorrelation Property

This transform has another important property which is used in further simplification of computations. The elements of the transformed vectors are maximally decorrelated. That is the correlation between  $i^{th}$  and  $j^{th}$  element of the output vector will have close to zero correlation.

### 4.3.3 Class Separability Measurement

After all the samples have been transformed into a  $\mathbb{F} \times C$  dimensional space  $\mathbb{I}$ , class scatter analysis is performed. In this space, the samples of each class can be thought of forming a cluster. We are interested in finding how the clusters are spaced, that

is the coherence of the cluster and the overlap that exists between clusters. We can notice over here that the variance of the class cluster can give us an idea about its scatter in the space giving an estimate of class coherence. Using the mean of each class in connection with variance, we can estimate the overlap that exists between classes.

Recalling the previous set up, there are  $M$  classes  $\{\Omega^{(m)}\}_1^M$  each having  $N_m$  number samples. Let us say the samples, after undergoing feature extraction and length normalization, are represented by  $x_{m,j}$ . The class mean  $\mu^{(m)}$  and variance  $\Sigma^{(m)}$  of each class as follows:

$$\mu^{(m)} = \frac{1}{N_m} \sum_{i=1}^{N_m} x_{m,i} \quad (7)$$

$$\Sigma^{(m)} = \frac{1}{N_m} \sum_{i=1}^{N_m} (x_{m,i} - \mu^{(m)})(x_{m,i} - \mu^{(m)})^T \quad (8)$$

For two classes  $\Omega^{(p)}$  and  $\Omega^{(q)}$ , the Bhattacharya measure is defined as,

$$D_B(p, q) = \frac{1}{4} (\mu^{(p)} - \mu^{(q)})^T [\Sigma^{(p)} + \Sigma^{(q)}]^{-1} (\mu^{(p)} - \mu^{(q)}) + \frac{1}{2} \log \left( \frac{|\Sigma^{(p)} + \Sigma^{(q)}|}{2(|\Sigma^{(p)}||\Sigma^{(q)}|)^{\frac{1}{2}}} \right) \quad (9)$$

The computation involves inversion of the covariance matrix which is still a costly operation. This can be decreased by making the covariance diagonal. If the elements of the sample vector  $x_{m,j}$  are uncorrelated, the corresponding covariance matrix will be near to diagonal. DCT based transformation is applied to feature vectors in the temporal direction, the output elements in that direction are uncorrelated. Also the base vectors are obtained from the cepstrum base feature extraction system which performs a DCT based transformation before producing the feature vector(Figure 4.5). This would imply that the correlation between the feature elements has been drastically reduced. Hence, only diagonal covariance matrix can be used instead of the full covariance matrix.

As our problem consists of more than two classes, we need to extend the two class measure to multi-class measure. This has been done in the literature[4] using the

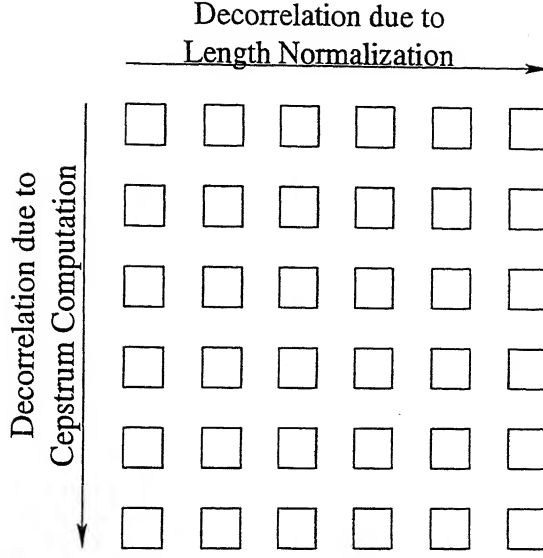


Figure 4.5: Decorrelation in measurement space

a priori probability of the classes considered. Let us say, the a priori probability of the class  $\Omega^{(m)}$  is  $Pr(\Omega^{(m)})$ . Then the distance measure for a multi-class measure can be given as,

$$D_{ave} = \sum_{i=1}^M \sum_{j=1}^M Pr(\Omega^{(i)}) Pr(\Omega^{(j)}) D_B(i, j) \quad (10)$$

In our case, we assume that all the classes are equally likely. Therefore the Equation 10 can be simplified using  $Pr(\Omega^{(i)}) = \frac{1}{M}$ . Hence the final equation would be,

$$D_{ave} = \frac{1}{M^2} \sum_{i=1}^M \sum_{j=1}^M D_B(i, j) \quad (11)$$

The average distance obtained using the above equation is used as the fitness measure for the filter-bank provided.

## Discussion

In this chapter, a GA-based optimization framework has been described using filter-bank based feature extraction as search space. It is possible to search another set of feature space induced by other feature extraction systems by changing the

genetic representation and by defining the operators. The objective function need not be changed as far as the feature extractor falls in to the frame-based category which is mostly the case.

# Chapter 5

## Experiments and Results

### 5.1 Baseline Recognition System: SPHINX

For testing purpose, the evolved filter-bank was tested using speech recognition. A complete speech recognition system is required which is modular enough to facilitate changes needed to incorporate the evolved feature extractor. The two open source system considered were SPHINX[25] and HTK[23]. SPHINX was chosen for our experiment purpose because of its modularity and flexibility. A brief overview of the training and decoding systems is provided in this section.

Sphinx is a set of HMM based speech recognition engines and training programs developed at Carnegie Melon University (CMU). It includes the decoding engines Sphinx-2, Sphinx-3 and Sphinx-4 and a set of training tools SphinxTrain. Of these systems, we are used Sphinx-4 as a decoding system because of its modularity and flexibility. It is specifically designed to provide a modular platform for speech recognition experiments. SphinxTrain is used to train the HMM models.

An overall architecture of the system is depicted in the Figure 5.1. Each labeled element is a configurable module that can be easily replaced allowing researchers to perform different experiments with different modules without changing other parts. Sphinx-4 provides both, simple and state-of-the-art implementations for each module. As with other speech recognition systems, Sphinx-4 has a large number

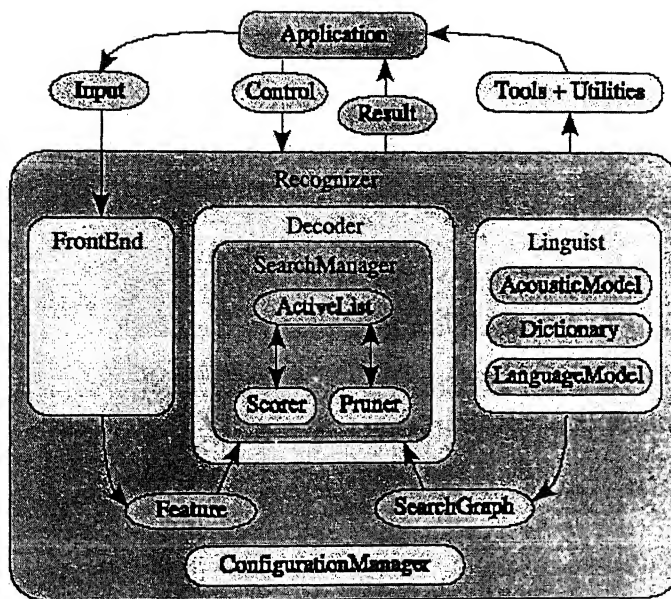


Figure 5.1: Sphinx - 4 Decoder Framework

of parameters pertaining to each module. Also there must be some way to specify which implementation has to be used for each module. A *Configuration Manager* is used to perform this task. A different implementation of a front-end can be placed into the system without changing any other code or even recompilation. It uses a global configuration file containing all the specification of the system. The framework is defined using Java Interfaces.

In our implementation, the front-end module is modified to experiment with evolved filter-banks. A default implementation is used for other modules. A detailed block diagram of the front-end is shown in Figure 5.2. It comprises one or more parallel chains of replaceable communicating modules called *DataProcessors*. This enables the system to use different kinds of feature extractors with different combinations. A frame-based feature extractor discussed in Chapter 3 is implemented in Sphinx as a chain of *DataProcessors*. In the filter-bank processing block, a Mel scale-based filter-bank is used. Hence, this block was modified to process the



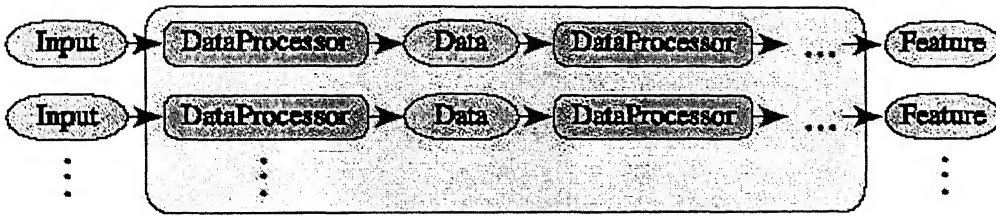


Figure 5.2: Sphinx-4 Front-end

input spectrum using any filter-bank specified. The specific filter-bank can now be specified in the configuration file.

## 5.2 Data Sets

As the method is data driven, the availability of specific type of data is very important. In our experiments, we have used three data sets.

### 1. *Prologix* Data Set

This data set contains phonetically tagged Hindi speech data from a single female speaker. A phonetically tagged speech data is one which contains a mapping between the speech signal and its transcription at the phoneme level. This is the main dataset used in our learning experiments as they are performed on phonemes.

Speech data can be phonetically tagged using the forced alignment[23] method following a manual check of the tagging. First the HMM models are trained on some of the speech data. The HMM models are built at the phoneme level. These models are then used with an untagged speech signal and its corresponding phonemic transcription to find the best match. This process is called forced alignment. The tagging generated by this process has some misalignments that have to be corrected manually. This data set was obtained from Prologix Software Pvt. Ltd.

## 2. *MLA* Data Set

This data set contains 44 Hindi words used in a general Pathology Lab Task. Each word is spoken by 12 speakers. This data is recorded in an unrestricted environment having different kinds of noise.

## 3. *Hindi* Data Set

This is also an isolated word data set containing 5500 Hindi words spoken by 7 male and 3 female speakers. Recording of this data was done in clean environment.

# 5.3 Hindi Phoneme Recognition Task

Phonemes are the speech units that are used in speech recognition systems as a base model. The acoustic models for higher level speech units are generally built combining the acoustic models at the phonemic level. As discussed in Chapter 2, the acoustic models for phonemes are generally built using HMMs. Hence our aim is to search for features that would enable better recognition at the phoneme level which would in turn increase the accuracy at higher levels.

Here the recognition task is to classify the Hindi Phoneme set. These are 48 phonemes including one silence phone making  $M = 48$ . Each class has 40 samples extracted from the tagged continuous speech data uttered in different contexts. To modify a filter-bank randomly in a small neighborhood in the parameter space, the variable *md*-maximum deviation is set to 7. That is the edge of the filter ( $\alpha, \beta, \gamma$ ) can be randomly changed between -7 to +7 frequency bins. The filter-bank consists of 40 triangular filters spaced between 133Hz to 6800Hz. The values of other parameters used in this experiment are shown in Table 5.1.

The result of the optimization and the corresponding accuracy is shown in Figure 5.3. The recognition accuracy is obtained using the *MLA* data set with 10 speakers used for training and 2 speakers used for testing. The dash-dotted line shows the fitness of the best individual in every population and the solid line shows the

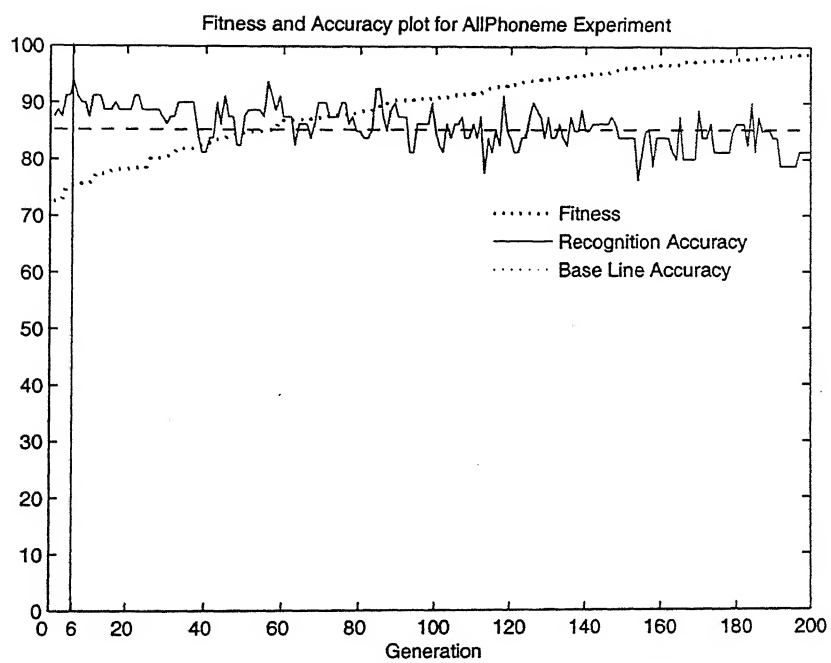


Figure 5.3: All Phoneme Experiment Results

Parameter Name	Value
Number of FFT points	512
Population Size	100
Mutation rate	0.1
Crossover rate	0.9
Number of Generations	200

Table 5.1: Parameter values for AllPhoneme Experiment

accuracy obtained on the *MLA* data using the filter-bank of the corresponding individual. The dotted line shows the accuracy obtained using Mel filter-bank (say  $FB_{mel}$ ) instead. The graph of the recognition accuracy shows oscillatory behavior with respect to fitness, though the initial half of the generations, it remains higher than  $FB_{mel}$ . Some of the filter-banks perform 5-7% better than the Mel Scale.

The envelope of the accuracy graph seems to be decreasing with increase in number of generations. This is interesting behavior which can potentially give a new method for speaker adaptation. As the GA training is performed using the speech data from a single speaker, the filter-banks are moving in a direction in the parameter space specific to that specific speaker. More experiments in this direction can provide a method which derives a filter-bank specific to a speaker. However, to develop a filter-bank for speaker independent recognition, this behavior is not desirable. The phonetically tagged data from a range of speakers can be used, instead to alleviate this problem. In that experiment, it is expected that the optimization would learn a filter-bank pattern which achieves high fitness throughout the range of speakers.

The filter-banks achieving high accuracy can be used in speech recognition systems if the improvement is sustained across different data sets. To check this property, the experiments were carried out on the *Hindi* data set. We have compared the baseline  $FB_{mel}$  and the filter-bank which is marked in the Figure 5.3, say  $FB_{allphoneme}$ . This filter-bank is generated at the sixth generation of evolution. The training was done on 5 speakers and testing was done on 2 speakers. Table 5.2 shows the comparison

of the performance of both filter-banks on MLA and Hindi data sets. We notice that the accuracy obtained using  $FB_{allphoneme}$  is higher than  $FB_{mel}$ .

Data Set	Number of Words	Recognition Accuracy	
		$FB_{mel}$	$FB_{allphoneme}$
<i>MLA</i>	44	85.22%	93.75%
<i>Hindi</i>	20	75%	82.25%
	40	81.25%	83.75%
	80	56.87%	65.62%

Table 5.2: Comparison of  $FB_{mel}$  and  $FB_{allphoneme}$  on *MLA* and *Hindi* data set

Figure 5.4 shows both the filter-banks. Most of the filters in these banks are similar except some filters have their left or right frequency edge displaced. Figure 5.5 shows the center frequencies of both the filter-banks.

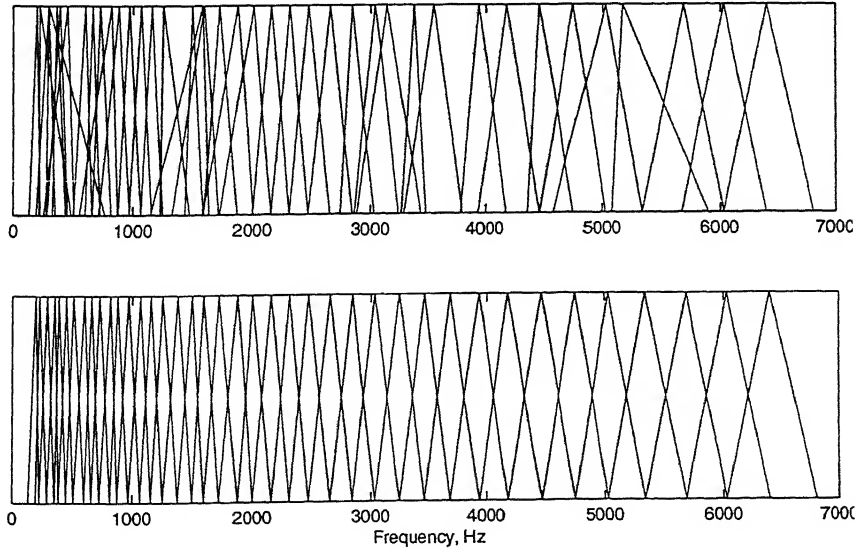


Figure 5.4:  $FB_{allphoneme}$  (top) and  $FB_{mel}$  (bottom)

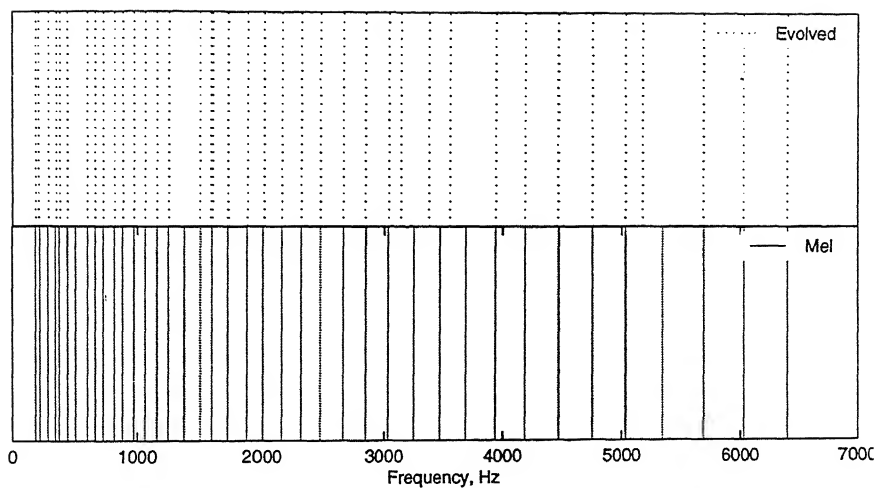


Figure 5.5: Center Frequencies of the Best Filter Bank

In another experiment, only the vowels in Hindi were used as a class definition including a silence phone. All the other parameters were kept same. Here also we have obtained similar behavior for the accuracy plot.

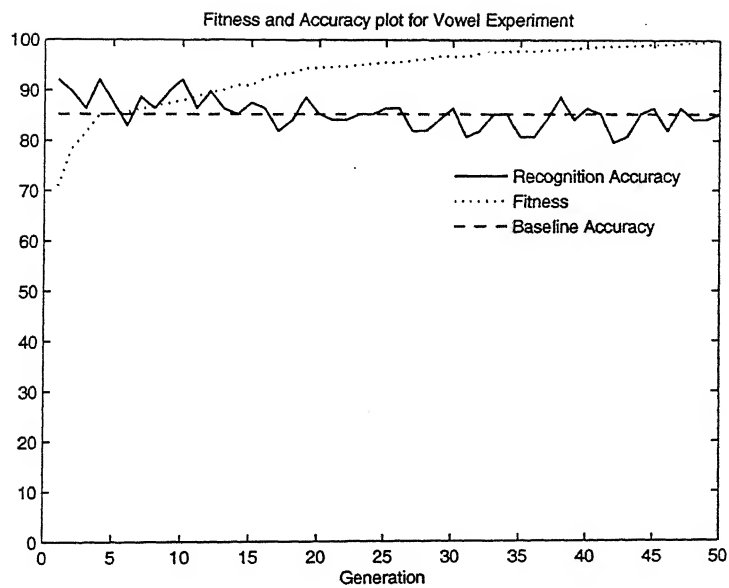


Figure 5.6: Vowel Experiment

# Chapter 6

## Conclusion and Future Work

### 6.1 Conclusion

In this thesis, we have attempted to obtain a feature extractor which retains the discriminative information through maximizing the class separability criteria. To the best of our knowledge, this is the first time genetic algorithms have been used for optimization of feature extraction for recognizing speech. The results of experiments indicate that GA is able to search the space obtaining better filter-banks in every generation with respect to a given Objective Function. This proves that the operators, mutation, crossover and initialization, have been defined properly to facilitate the search algorithm to search the space efficiently. Genetic algorithms can be used in such optimization without having any constraints on the objective function. Hence more complex objective functions can be used.

From the experiments performed on Hindi vowels and All Hindi Phonemes, some filter-banks have been obtained which perform significantly better in terms of accuracy than the baseline Mel systems. The performance improvement has been observed on two data sets, MLA dataset and Hindi dataset. This shows that there exist filter-banks which can give better results. Analysis of these filter-banks, the spread of the filters and their band widths, can provide interesting information about the critical energy bands required. These evolved filter-banks can be used in Hindi



speech recognition systems to get the performance improvement.

## 6.2 Future Work

Current work gives indications that more exploration in filter-bank based feature extraction can lead to better feature extractors. The effect of noise on different kinds of feature extractors has been well studied in literature. Generally, this work is focused on changing the filter-bank bandwidths[20] to get noise robustness. If the GA based approach is applied to noisy data, the obtained filter-bank is expected to be robust to that kind of noise. The other experiment that can be done is to initialize the filter-banks with different strategies like random initialization, nearly linear initialization, mixture of variations of Mel and Bark scale filter-banks. The path that the filter-banks traverse and its convergence can give interesting information about the filter-banks being used.

The aspect of memory requirement can also be studied. That is to see how a learned filter-bank performs with reduced Gaussians and states per speech unit. The objective function used in our work is giving an oscillatory behavior. More exploration on the properties of objective function and its behavior can lead to monotonic improvement in the recognition accuracy.

### Optimization in other search spaces

The optimization framework is able to search for the best feature extraction parameters in the search space defined by a specific feature extraction method. In our work, we have used Fourier Transform Frame-based feature extraction. Other potential method which can be parameterized is based on wavelet transform. A wavelet transform is used for multi-resolution analysis of the signal where the *stationary* property of the signal is not required. Therefore this transform can parameterize the speech signal without having to assume it is stationary. Also the speech contains phonetic information at different resolutions. Hence a wavelet-based parameterization is expected to provide better features. The improvement in the accuracy has already been seen in [11].

# Bibliography

- [1] AGRAWAL, R., FALOUTSOS, C., AND SWAMI, A. N. Efficient similarity search in sequence databases. In *International Conference on Foundations of Data Organization and Algorithms* (1993), pp. 69–84.
- [2] BIEM, A. E. *Discriminative Feature Extraction Applied to Speech Recognition*. PhD thesis, University of Paris, 1997.
- [3] BIEM, A. E. Discriminative feature extraction applied to speech recognition. In *IEEE Transactions on Acoustics, Speech, and Signal Processing* (February 2001), vol. 9, pp. 96–108.
- [4] BRUZZONE, L., AND SERPICO, S. B. A technique for feature selection in multiclass problems. *International Journal of Remote Sensing* 21, 3 (2000), 549–563.
- [5] BUCHSBAUM, A. L., AND GIANCARLO, R. Algorithmic aspects in speech recognition: An introduction. *Journal of Experimental Algorithmics* 2, 1 (1997).
- [6] DE LA TORRE, A., PEINADO, A. M., RUBIO, A. J., AND GARCAI, P. Discriminative feature extraction for speech recognition in noise. In *Proceedings of EUROSPEECH* (97).
- [7] DEVIS, S. B., AND MERMELSTEIN, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28, 4 (August 1980).
- [8] DUDA, R. O., HART, P. E., AND STORK, D. G. *Pattern Classification*. John Wiley & Sons, Inc, 2002.

- [9] FALOUTOS, C., RANGANATHAN, M., AND MANOLOPOULOS, Y. Fast subsequence matching in time-series databases. In *In Proceedings of the ACM SIGMOD International Conference on Management of Data* (May 1994), pp. 419–429.
- [10] FANT, C. G. M. Acoustic description and classification of phonetic units. *Speech Sound and Features* 15, 1 (1973).
- [11] FAROOQ, O., AND DATTA, S. Wavelet based robust sub-band features for phoneme recognition. In *IEE Proc.-Vis. Image Signal Process.*, (June 2004), vol. 151, pp. 187–193.
- [12] FUKUNAGA, K. *Introduction to Statistical Pattern Recognition*. Academic Press, 1972, ch. 9.
- [13] LEE, C.-H., SOONG, F. K., AND PALIWAL, K. K., Eds. *Automatic Speech and Speaker Recognition Advanced Topics*. Kluwer Academic Publisher, 1995.
- [14] LIU, X. Linear projection schemes for automatic speech recognition. Master's thesis, University of Cambridge, August 2001.
- [15] MALAYATH, N., AND HERMANSKY, H. Data-driven spectral basis functions for automatic speech recognition. *Speech Communication* 40 (2003), 449–466.
- [16] POPIVANOV, I., AND MILLER, R. Similarity search over time-series data using wavelets. In *Proceedings. 18th International Conference on Data Engineering* (Feb 2002), pp. 212–221.
- [17] RABINER, L., ROSENBERG, A. E., AND LEVINSON, S. E. Considerations in dynamic time warping algorithm for discrete word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26, 6 (December 1978), 575–582.
- [18] RABINER, L. R. A tutorial on hidden markov models and selected applications in speech recognition. In *In Proceedings of the IEEE* (1989), vol. 77, pp. 257–285.

- [19] SHIGERU, K., Ed. *Handbook of Neural Networks for Speech Processing*. Artech House Publishers, Boston, London, 2000.
- [20] SKOWRONSKI, M. D. Improving the filter bank of a classic speech feature extraction algorithm. In *IEEE Intl Symposium on Circuits and Systems* (May 2003), vol. IV, pp. 281–284.
- [21] SKOWRONSKI, M. D. *Biologically inspired noise-robust speech recognition for both man and machine*. PhD thesis, UNIVERSITY OF FLORIDA, 2004.
- [22] SLANEY, M. Auditory toolbox. Tech. Rep. 1998-010, Interval Research Corporation, 1998.
- [23] STEVE YOUNG, G. E. *The HTK book*, 3.2 ed. Cambridge University University Department, December 2002.
- [24] TRENTIN, E., AND GOR, M. A survey of hybrid ann/hmm models for automatic speech recognition. *Neurocomputing* 37, 4 (April 2001), 91–126.
- [25] WALKER, W., LAMERE, P., KWOK, P., RAJ, B., SINGH, R., GOUVEA, E., WOLF, P., AND WOELFEL, J. Sphinx-4: A flexible open source framework for speech recognition. Tech. Rep. SMLI TR2004-0811, SUN MICROSYSTEMS INC., 2004.
- [26] WALLACE, G. K. The jpeg still picture compression standard. *Communications of the ACM*. (April 1991).
- [27] ZWEIG, G. G. *Speech Recognition with Dynamic Bayesian Networks*. PhD thesis, University of California, Berkeley, 1998.